VMware® ThinApp™

REVIEWER'S GUIDE

vmware®

Table of Contents

Technology Overview . . . 3
- Common Use Cases to Leverage VMware ThinApp . . . 3
- Review of Key Features . . . 3

Getting Started with VMware ThinApp . . . 5

ThinApp Packaging Process . . . 6
- Packaging an Application Using Setup Capture . . . 6
- Using Active Directory groups to authorize ThinApp Packages . . . 7
- Modifying Settings in the Package.ini File . . . 8
- Harvesting Internet Explorer 6 and Using ThinDirect with Setup Capture . . . 9
- Using ThinApp Converter for Automated Packaging . . . 10

Deploying ThinApp Packages . . . 11
- ThinApp Deployment with VMware View 4.5 . . . 11
- Centralized and De-Centralized Deployment . . . 11
- Application Registration . . . 13
- Role-Based Access to Applications . . . 13
- Deploying ThinApp Packages . . . 14

Using ThinApp Assignments with VMware View . . . 15
- To Create the MSI-based Packages for Full and Streaming Deployment . . . 15
- To Create the ThinApp Repository and Populate With ThinApp Packages . . . 16
- To Create ThinApp Assignments in View Administrator . . . 18
- To Utilize Script-based Application Registration . . . 19
- To Utilize AppLink to Combine ThinApp Packages . . . 20

Updating ThinApp Packages . . . 20
- Packaging Updates and Modifications . . . 20
- Deploying Updates . . . 21
- To Update ThinApp Packages Using the Side by Side Method . . . 22
- To Utilize AppSync to Update ThinApp Packages . . . 22

Additional Resources . . . 23

About the Author . . . 23

Technology Overview

ThinApp is an agentless application virtualization solution that decouples applications from their underlying operating systems to eliminate application conflict and streamline application delivery and management. VMware ThinApp™ simplifies application virtualization to enable IT administrators to quickly deploy, efficiently manage, and upgrade applications without risk, accelerating desktop value. With ThinApp, an entire application and its settings can be packaged into a single executable and deployed to many different Windows operating systems without imposing additional cost and complexity to the server or client. Application virtualization with ThinApp eliminates conflicts at the application and OS level and minimizes costly recoding and regression testing to speed application migration to Windows 7.

ThinApp virtualizes applications by encapsulating application files and registry into a single ThinApp package that can be deployed, managed, and updated independently from the underlying operating system (OS). The virtualized applications do not make any changes to the underlying OS and continue to behave the same across different configurations for compatibility, consistent end-user experiences, and ease of management.

As a key component of VMware View, ThinApp adds application compatibility to virtual desktop environments and helps reduce the management burden of applications and desktop image management. IT Administrators have the benefit of a single console to deploy and manage desktops and applications using the integrated functionality of assigning ThinApp packages within the View Administrator console.

Common Use Cases To Leverage VMware ThinApp

VMware ThinApp simplifies application delivery by encapsulating applications in portable packages that can be deployed to many end point devices while isolating applications from each other and the underlying operating system.

- Simplify Windows 7 migration—Migrate legacy applications such as Internet Explorer 6 to 32- and 64-bit Windows 7 systems with ease by using ThinApp to eliminate costly recoding, regression testing, and support costs.
- Eliminate application conflicts—Isolate desktop applications from each other and from the underlying OS to avoid conflicts.
- Consolidate application streaming servers—Enable multiple applications and “sandboxed” user-specific configuration data and settings to safely reside on the same server.
- Reduce desktop storage costs—Leverage ThinApp as a component of VMware View to reduce desktop storage costs and streamline updates to endpoints.
- Augment security policies—Deploy ThinApp packages on “locked-down” PCs and allow end users to run their favorite applications without compromising security.
- Increase mobility for end users—Deploy, maintain, and update virtualized applications on USB sticks for ultimate portability.

Review of Key Features

Agentless Application Virtualization

- Agentless architecture—Designed for fast deployment and ease of management, ThinApp requires no agent code on target devices.
- Complete application isolation—Package entire applications and their settings into a single executable that runs independently on any endpoint, allowing multiple versions or multiple applications to run on the same device without any conflict.

- Built-in security—Application packages run only in user mode, so end users have the freedom and flexibility to run their preferred applications on locked-down PCs without compromising security.

Fast, Flexible Application Packaging

- Package once, deploy to many—Package an application once and deploy it to desktops or servers (physical or virtual, 32- or 64-bit) running Windows XP, Windows Vista, Windows 7, Windows Server 2003, or Windows Server 2008.
- Three-step setup capture—Use a three-step process for pre and post-install system states to simplify application packaging and support applications that require a reboot during the installation process.
- Microsoft Internet Explorer 6 support—ThinApp now offers complete support for virtualizing Microsoft Internet Explorer 6 (IE 6) that makes it easy to virtualize and deploy IE 6 application packages to 32- and 64-bit Windows 7 desktops.
- ThinApp Converter—ThinApp works with VMware vSphere™, VMware ESX®, and VMware Workstation images to convert silently installed applications into ThinApp packages through a command-line interface that allows for automation of application conversion and management.
- Relink—Upgrade existing ThinApp packages to the new ThinApp 4.6 format quickly and easily without the need for associated project files.

Fast, Flexible Application Delivery

- ThinDirect—This new feature gives end users the flexibility to seamlessly run IE 6 on Windows 7 desktops alongside newer browsers such as IE 8, and allows the administrator to configure Web pages with IE 6 dependencies to ensure that URLs always open in the right browser.
- Application link—Configure relationships between virtualized applications, plug-ins, service packs, and even runtime environments such as Java and .NET.
- Application sync—Automatically apply updates over the Web to applications on unmanaged PCs and devices.
- Support for USB drives and thin clients—Deploy, maintain, and update applications on USB storage drives and thin client terminals.
- Microsoft Windows 7 support—Deploy legacy applications on 32- and 64-bit Windows 7 systems and streamline application migration by avoiding costly, time-consuming recoding and regression testing.

Seamless Integration with Existing infrastructure

- Zero-footprint architecture—Plug ThinApp directly into existing IT tools without the need to add dedicated hardware or backend databases.
- Integration with management tools—ThinApp creates standard .MSI and EXE packages that can be delivered through existing tools from Microsoft, BMC, HP, CA, Novell, Symantec, LANDesk, and others.
- Support for Active Directory authentication—Add and remove ThinApp users from Active Directory groups, and prevent unauthorized users from executing ThinApp packages.
- Integrated application assignment in VMware View Manager 4.5—ThinApp packages can be assigned to individual desktops or pools of desktops in View Manager to allow for streamlined application deployment.

Getting Started with VMware ThinApp

What you will need to get started:

1. VMware ThinApp software and product key
2. Pristine installation of a Windows OS
3. Application install source files
4. VMware View 4.5 environment (optional)

To obtain the evaluation software and product key for 50 clients go to http://www.vmware.com/go/trythinapp

Your trial includes:

- VMware ThinApp Packager & 50 Client licenses
- VMware Workstation

Additional information can be found at the ThinApp Community website - http://communities.vmware.com/community/vmtn/entdesk/thinapp

HTML based Product Documentation is available here - http://pubs.vmware.com/thinapp4/help/wwhelp/wwhimpl/js/html/wwhelp.htm

Supported platforms:

ThinApp supports various operating systems, applications, and systems.

- 32-bit platforms include Windows NT, Windows 2000, Windows XP, Windows XPE, Windows 2003 Server, Windows Vista, Windows Server 2008, Windows 7
- 64-bit platforms include Windows XP 64-bit, Windows 2003 64-bit, Windows Vista 64-bit, Windows Server 2008 64-bit, Windows Server 2008 R2 64-bit, Windows 7 64-bit
- 16-bit applications running on 32-bit Windows operating systems
- 32-bit applications running on 32-bit and 64-bit Windows operating systems

Not supported:

- 16-bit or non x86 platforms, such as Windows CE
- 64-bit applications
- Applications requiring installation of kernel-mode device drivers (ODBC drivers work because they are user-mode)
- Products such as anti-virus and personal firewalls
- Scanner and printer drivers, some VPN clients

ThinApp Packaging Process

The process of virtualizing an application with VMware ThinApp begins with the setup capture process and ends with the build of a read-only redistributable package that encapsulates all the necessary components of the application along with the administrator supplied configuration settings necessary for implementation. The setup capture process creates a project to store the application and configuration settings. The build process compresses and embeds the project directories and configuration settings into the package. The project directory is the source location where the administrator can return to make subsequent updates or changes to the package configuration. The result of making a configuration change and rebuilding would be two separate packages created from the same VMware ThinApp project but with different configuration settings. The process of using setup capture, the project directories, and the build process is meant to be an iterative process, often referred to as 'Capture and Build.'

While there are distinct operations in these phases, it is helpful for the administrator to be mindful of the future deployment model when using the setup capture process to package the application. Configuration settings that describe the update method, specific application characteristics, and integration with the local operating system are embedded into the package during the Capture and Build phases. Building the package is the logical transition point between creating and deploying the application. The subsequent phases of Deploy and Update utilize the package as a modular application container, which is then distributed and updated accordingly. It is important to note that the packaging process can produce an MSI package in addition to the default EXE based package.

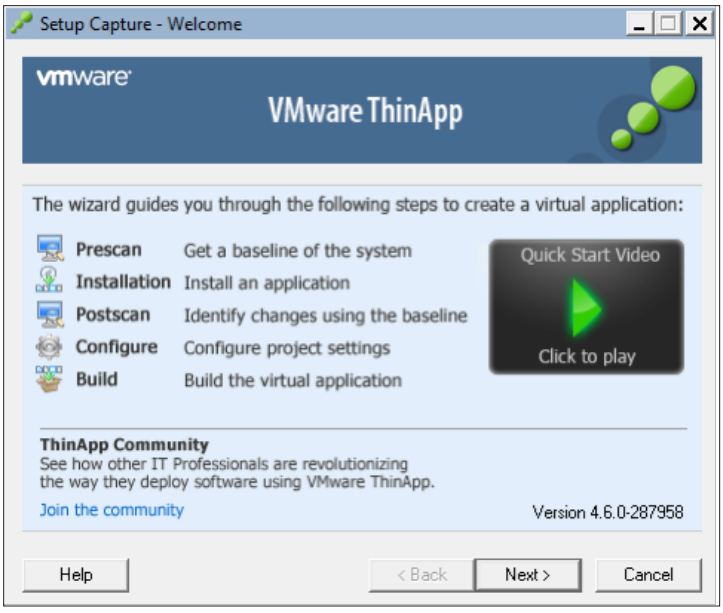
Product documentation for the Setup Capture process can be found here: http://pubs.vmware.com/thinapp4/help/wwhelp/wwhimpl/common/html/wwhelp.htm?context=help&file=capturing_apps.html#1051933

Packaging an Application Using Setup Capture

Install the VMware ThinApp Software via the MSI onto a clean operating system or place files on a network share accessible from the setup capture operating system instance.

Go to Start > Programs > VMware > ThinApp Setup Capture or run the Setup Capture from a mapped network location that houses all of the VMware ThinApp files.

Watch the Quick Start Video and utilize the contextual help for detailed guidance throughout the process.



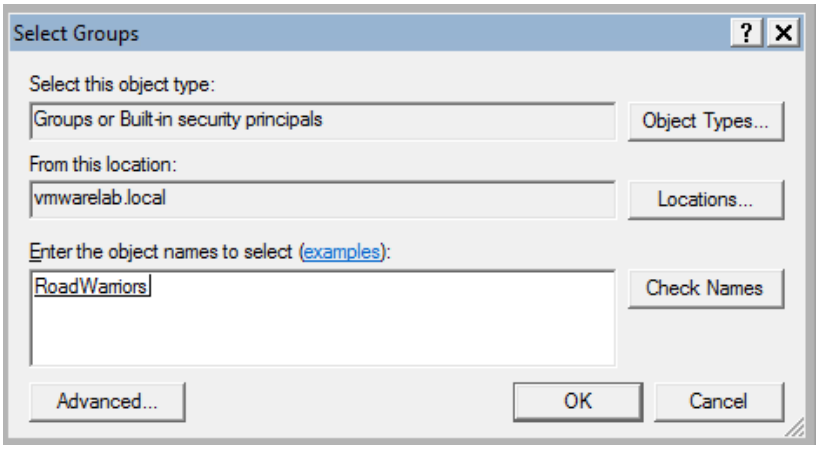
Note: Use a Clean Operating System for Setup Capture

VMware recommends that customers install only the basic components of the operating system for the machine that is used for Setup Capture. It is also recommended that you use the oldest version of the operating system to ensure that the application install puts down all required files. The reason to capture from a clean operating system is to ensure that all the files and components necessary for the application are detected by the Setup Capture process. If there is application install logic that looks for a certain version of a .dll and the capture machine finds it in the local operating system, then that .dll will not be included in the virtualized application package, and that may not allow the application to function when deployed to an operating system instance that has not been updated.

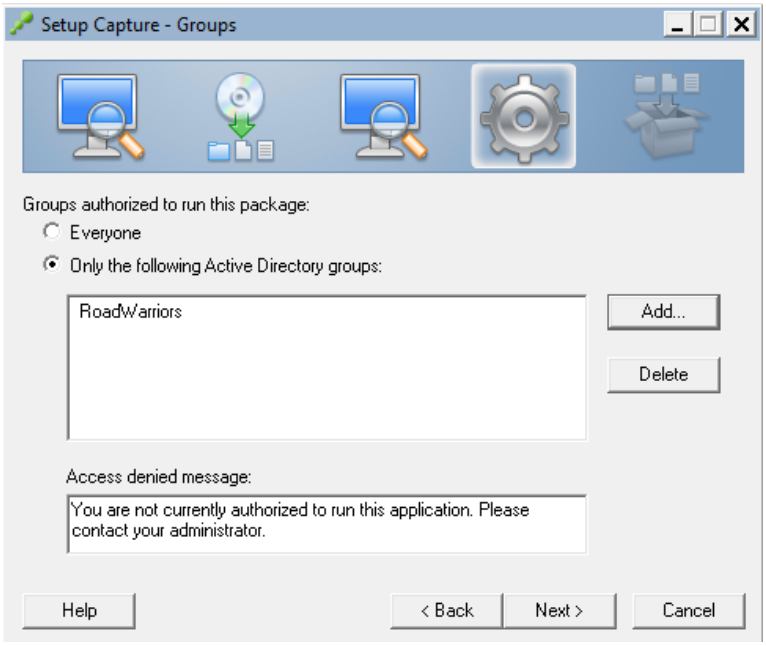
Using Active Directory groups to authorize ThinApp Packages

The process of deploying virtualized applications offers administrators control and flexibility over which machines and users receive the application packages and the ability to launch the packages. Utilizing Active Directory allows an organization to use the existing processes and controls for group-based security. In addition to these organizational controls, VMware ThinApp allows an administrator to embed access control into the package. This access control mechanism is obfuscated from the end user when the package is built so it is impossible to identify or remove before the application is launched. In this way, the access control travels with the package if it is moved between devices after deployment.

1. Select the Active Directory groups that you want to authorize for access to the application.
2. Click **Add**
   - To specify objects, click **Object Types**.
   - To specify a location in the forest, click **Locations**.
   - To search object names, enter the names according to the examples in the dialog box.
3. To locate user names in the Active Directory forest, click **Advanced** and use the **Common Queries** tab to search for groups according to names, descriptions, disabled accounts, passwords, and days since the last login.

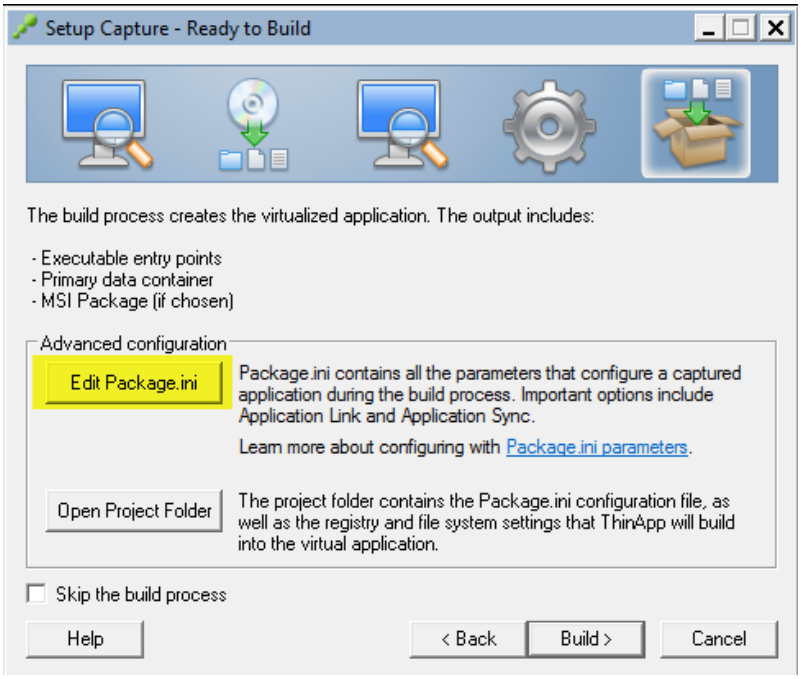


The following example shows that an administrator has selected 'RoadWarriors' as the only Active Directory group able to launch this application.



Modifying Settings in the Package.ini File

The last step of the Setup Capture Wizard prompts the administrator for advanced configuration before commencing 'Build' of the package. The **Package.ini** file contains configuration settings and resides in the captured application folder.



1. Click Edit Package.ini
2. Modify a parameter by removing the semicolon at the beginning of the line or editing existing parameters.
3. For example, activate the RemoveSandboxOnExit parameter by deleting the semicolon at the beginning of the line so that RemoveSandboxOnExit=1 is enabled in the package.ini.
4. Modify any additional parameters and save the file.
5. Return to the Setup Capture Wizard and click 'Build' to complete the process and generate ThinApp packages.

Subsequent sections will refer to several package.ini modifications. When making changes to these parameters the change is first made to the package.ini, then the packages must be rebuilt to embed the administratively configured settings within the package. Rebuilding packages can be done at anytime by browsing to the Project Folder and running the Build.bat file.

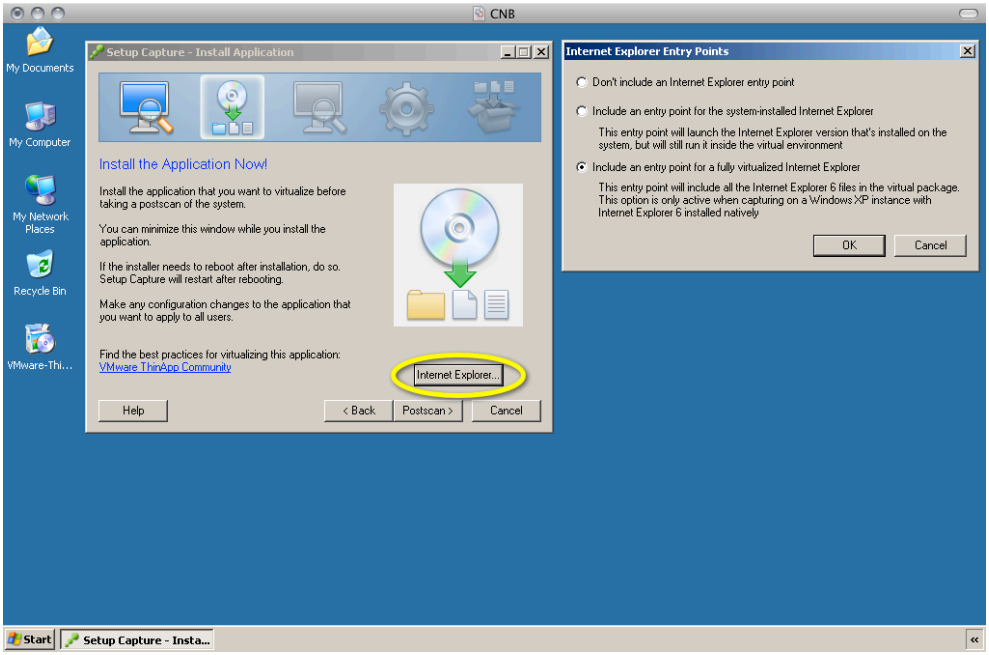
Harvesting Internet Explorer 6 and Using ThinDirect with Setup Capture

VMware ThinApp 4.6 allows customers to Capture IE 6 on Windows XP with easy click-through packaging, full rendering to display IE 6 application web pages, the ability to run virtualized IE 6 and natively installed IE 7 and IE 8 seamlessly in the same desktop, compatibility for all system shell commands, and 32-bit and 64-bit support.

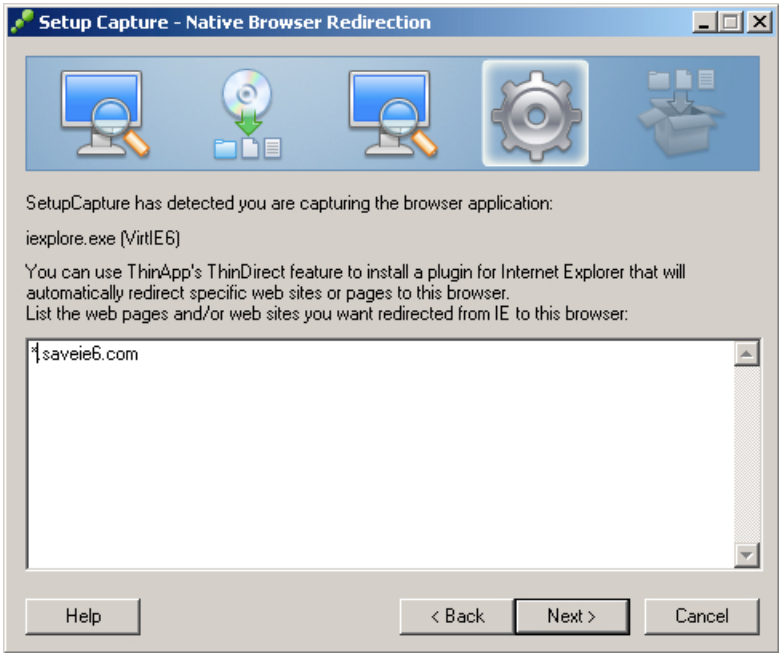
Harvesting is a method of extracting the existing Internet Explorer 6 from the Windows XP operating system without actually installing the application. The following procedure outlines the steps to do this.

See the following KB article for more detailed information: http://kb.vmware.com/kb/1026565

1. Important: To utilize the harvest method of capturing Internet Explorer 6 you must run Setup Capture on a Windows XP operating system instance.
2. Run the Setup Capture and click the Internet Explorer button within the Setup Capture window to 'harvest' IE6 then select 'Include an Entry Point for a fully virtualized Internet Explorer'.



3. Add any browser plug-ins or modifications, then click PostScan.
4. Advance through the Entry Points, Groups, Isolation, Sandbox, and Quality Assurance dialogs. At this point you'll be presented with a dialog box that will automatically redirect users going to specified hosts or URLs into the virtualized IE 6 instance. These entries populate the ThinDirect.txt file and can be edited manually afterward. This can also be done dynamically on the end users workstations via Active Directory GPO's using the ThinDirect.ADM provided in the VMware ThinApp installation files. See the following blog post for details: http://blogs.vmware.com/thinapp/2010/08/vmware-thinapp-46-whats-new.html#more



5. Complete the Setup Capture Wizard and click 'Build' to complete the process and generate ThinApp packages.

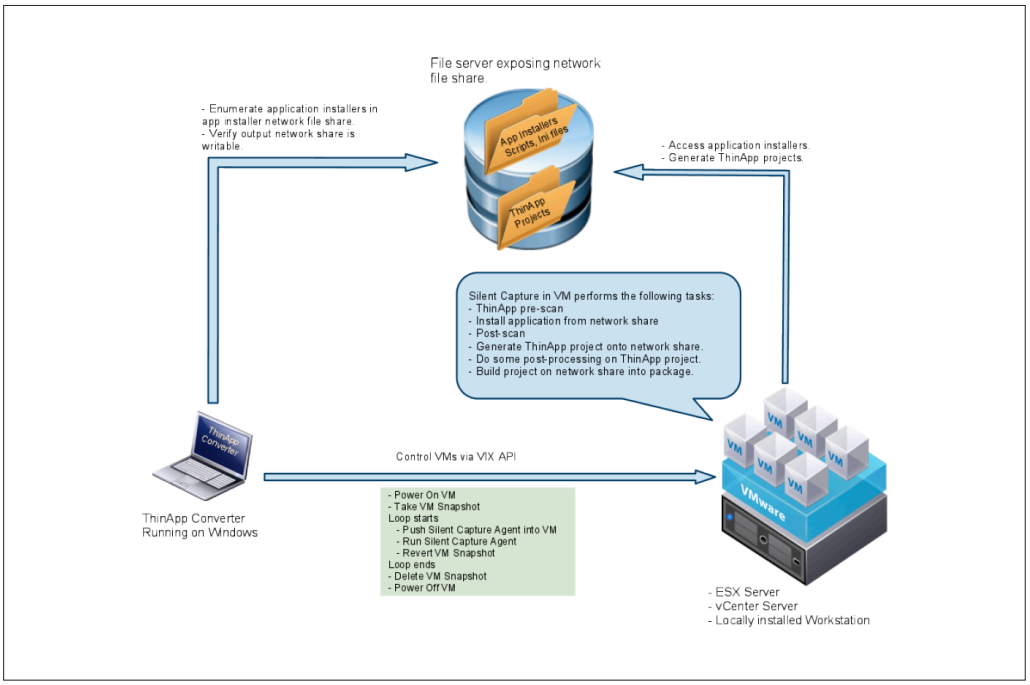
This procedure will result in an IE6 package as an EXE or MSI package that can be deployed to Windows XP or Windows 7 and run in parallel with other virtualized browsers or native browsers. The ThinDirect functionality gives administrators the flexibility to seamlessly redirect end users into either virtual or native browsers based on a whitelisting model.

Utilizing ThinApp Converter for Automated Packaging

VMware ThinApp 4.6 allows customers to automate the process of packaging applications and utilize virtual machines on VMware Workstation 7.0 and Virtual Infrastructure using ESX 4 or through vCenter 4.0. ThinApp Converter allows the administrator to point at a source directory of the application installer's or MSI packages and then pre-configure the setup strings used and package.ini settings as overrides for the resulting package directory.

See the following documentation for more detailed information: http://pubs.vmware.com/thinapp4/help/capture_multiple_apps_using_converter.html#1051355

To utilize ThinApp Converter, you must specify the parameters used for the automation in the ThinAppConverter.ini file and then reference the appropriate fileshares. This diagram represents the components and process used for the packaging operation.



Deploying ThinApp Packages

The process for deploying ThinApp application containers is very simple as there is no actual installation of the application and interoperation with the local operating system is minimal. Deployment involves making a decision for a centralized or de-centralized model for application delivery and then integrating the virtualized applications into the desktop for end user accessibility.

ThinApp Deployment with VMware View 4.5

The following sections discuss both deployment and application registration with processes that can be utilized on physical and virtual desktops. With the release of VMware View 4.5, ThinApp packages can be deployed from within the View Administrator console and automatically registered. This turnkey integration of View desktops and virtualized applications offer tremendous efficiency and operational savings for VMware View environments. For VMware View environments we suggest you read through the concepts of execution mode and application registration to understand what happens automatically when you assign ThinApp packages through the View Administrator console. The specific steps to deploy ThinApp packages in VMware View environments are covered in the section Using ThinApp Assignments with VMware View.

Centralized and De-Centralized Deployment

The determination of centralized versus de-centralized depends on the execution mode of the package. The same package can be executed using either method but delivers different benefits and administration models. Execution Mode describes the two options for providing virtualized applications to end users.

- Streamed execution mode allows the application to be centrally stored and accessed by multiple users. Streamed execution mode is a one-to-many model which provides centralized deployment and update of an application package to multiple end users for execution via a Windows desktop shortcut.

- Deployed execution mode application packages are first deployed to the end users system, and accessed from the local device. Users execute the application from an application package that is now local, which allows for offline application use.

Choosing an Execution Mode

One of the decision points for virtualizing applications with VMware ThinApp is to choose which delivery model is appropriate for user groups and applications. There are two primary options of application delivery: 1) the streamed mode of execution, and 2) the deployed mode. Both of these options have requirements and benefits that are listed below. A hybrid approach is acceptable as well. Determine the appropriate execution mode for each application and user group.

Streamed Execution Mode

The streamed mode of execution will often be the best option for environments that are centralized and desktops are always online. In streamed mode, the application is launched from a shortcut on the start menu or desktop on the local workstation and then streamed into memory as the application requests files and registry. The details of ThinApp Streaming were discussed earlier in this guide.

Requirements

The user must always have access to the central network location where the applications reside.

Recommendations

The storage location that hosts the applications should be made highly available such that the physical uptime of either the host or storage device does not impact the environment. The use of any number of SAN, DFS, or file-replication technologies is sufficient to accomplish the objective of making the file share highly available and redundant.

The path through the network between the client device and the central network location should be robust. A virtualized application utilizes standard SMB protocol. The amount of network traffic will vary based on the application and the functions used by the end user.

Benefits

Centralized administration is the primary benefit of the streamed mode of execution. The 'one-to-many' model of providing an application to one location for many users provides an efficient and effective model for application delivery. Providing access to the application merely involves placing a shortcut to the application on the desktop and can be automated through the use of the Thinreg utility in a login script.

The application packages, which can be large in size, do not have to be delivered to the end user devices so there is no need to transfer large files across a network or integrate with a deployment mechanism to distribute them. Additionally, there is no local disk footprint on the end user device because the applications are streamed into memory.

For users that access applications from multiple devices, the streamed model of execution provides a single point of administration and a consistent user experience across multiple devices.

Deployed Execution Mode

Deployed Execution Mode involves distributing the virtualized application packages to the end user's operating system. The actual location of the package can be on the local file system or a USB device. In this distributed model, each client device receives the package locally and therefore can run the application regardless of network connectivity. End user devices that are occasionally or always offline will require this deployed execution mode.

Requirements

Distribution of the packages to the local operating system is required in this model. A number of options exist to fulfill this requirement: Active Directory-based publishing via Group Policy, 3rd party software deployment solutions, and/or custom scripted mechanisms. Users who are occasionally offline must have all applications and components deployed before working offline. Subsequent application deployment and updates are subject to network availability or a delayed update tactic such as providing CD's or USB devices with updates.

Recommendations
Integrate the delivery of packages, which can be large .exe or .msi files, with your existing organizational process. An existing process, such as Active Directory publishing via Group Policy, will have an already established support structure and administration workflow. You can use Group Policy to deploy software to groups, OU's, or individuals. See the following KB article for details: http://support.microsoft.com/kb/324750

Benefits
Once the application package is delivered, application performance and availability is not subject to network or storage dependencies.

Application Registration

The process of application registration integrates the virtualized application packages with the desktop operating system. The process of registering applications will take into account the access control mechanisms that allow administrators to restrict usage to specified Active Directory groups providing end users role-based access to virtualized applications. Application registration is not mandatory, ThinApp packages will launch and execute without registration. However, end users and administrators can benefit from the integration of virtualized applications into the desktop because of the following:

- Creation of Windows Shortcuts on the **Start Menu** and **Desktop** for the applications
- Creation of File Type Associations, Protocols, and Object Types to launch virtualized applications
- Creation of Application Entries in the **Add/Remove Programs** Control Panel applet

VMware ThinApp allows IT organizations to determine whether to use streaming or deployed execution mode or to adopt a hybrid approach that lets them manage a standard set of applications centrally while distributing others in deployed mode. The same virtualized application packages can be used for either execution mode. The application registration process performs the exact same function whether packages are local or remote. As described later, MSI packages use ThinReg to automatically perform application registration with the MSI installer. So the application registration entries are consistent regardless of the means use to perform the actual registration or where the package resides.

Registration of virtualized applications to end users creates shortcuts on the desktop, file-type, protocol, and object type associations, and entries in the Add/Remove programs applet of the Control Panel. The Thinreg.exe tool is a simple utility, which automates the registration process and can be run from any scripting mechanism or command prompt. The ThinReg.exe can be local to the operating system, on a remote share. An example would be placing the ThinReg tool in the netlogon share and calling it from a login script. Administrators can run ThinReg against an entire directory of ThinApp packages by using an asterisk (*) as a wildcard character.

Role-Based Access to Applications

Since the registration process can enumerate which users have access to application packages, the process can be run against an entire directory of application packages; however, it only registers the applications to which the user is entitled. Administrators can use pre-made scripts that run based on group membership to only register packages that are valid for a certain group or for individuals. Two common methods of implementation using Active Directory are described briefly below.

Script-based Registration
Implementation of the Thinreg executable can be incorporated into an existing login script with standard methods such as .bat, WSH, KIX, or vbScript. See example below:

```
%logonserver%\netlogon\thinreg.exe /Q \\company.com\applications\*.exe
```

Local Script via Registry Run Key or Active Directory GPO managed Login and Logout Script
IT organizations can choose to implement the application registration process locally on the workstations

instead of incorporating it into the login script. The Run key of the registry can call the Thinreg.exe file to perform the necessary functions on login. Placing the Thinreg.exe in the Windows directory simplifies the execution of the script and requires nothing more than the executable to function.

MSI-based Registration

Deployment of Virtualized Applications with Electronic Software Distribution (ESD) Tools and Active Directory

Organizations can integrate the delivery of ThinApp packages run in deployed mode whether they are .EXE or MSI based packages. These delivery mechanisms often have an already established support structure and administration workflow. You can use native Active Directory based Group Policy to publish or assign MSI packages to groups, OUs, or individuals. See the following knowledge base (KB) article for details: http://support.microsoft.com/kb/324750

An organization with an established mechanism for deploying MSI files, such as Active Directory, can deploy ThinApp MSI packages in the same manner that they would deploy native applications. The process of registering applications to the desktop makes use of the ThinReg utility whether the package is deployed as an .Exe based package or in an MSI wrapper. MSI-based packages are always 'installed' into the local operating system. However, ThinApp MSI packages actually contain the EXE based package and the ThinReg utility. The use of ThinApp MSI packages doesn't actually install anything but rather puts the application registration process into the MSI install. In summary, for organizations that use MSI-based ThinApp packages there will be no need to make use of the ThinReg utility to perform application registration.

Deploying ThinApp Packages

The last step of the Setup Capture Wizard prompts the administrator for advanced configuration before commencing 'Build' of the package. Once the build process has completed the completed ThinApp packages are found in the 'Bin' folder of the project directory. If both an EXE and MSI package was created then both types will be available. The packages contain all of the configuration information specified during the Setup Capture process. To deploy the packages simply copy or distribute them to the appropriate locations. Portability of these packages gives administrators and end users significant flexibility for distribution, use the Active Directory authorization mechanism to secure these packages if necessary.

To utilize deployed mode execution

1. Copy the EXE or MSI based package to a local file system or USB drive.
   Note: For the MSI package, you must double-click to install the package.
2. Launch the application and test functionality.

To utilize streamed mode execution

1. Copy the EXE based package to a remote fileshare
2. Drag a shortcut from the package to your desktop.
3. Launch the application and test functionality.

To utilize packages on multiple operating systems

1. Copy the EXE based package to a Windows XP, Windows Vista, Windows 7, Windows Server 2003
2. Publish the application as a Remote App using Microsoft Remote Desktop Services
3. Publish the application from a Citrix XenApp or Presentation Server
4. Launch the applications and test functionality.

To deploy MSI packages using Active Directory or 3rd Party ESD
Utilize the following blog article for detailed considerations of GPO deployment: http://blogs.vmware.com/thinapp/2009/03/notes-and-considerations-on-deploying-thinapp-packaged-applications-via-active-directory-group-policies.html

1. Create a GPO to assign an application
2. Select the MSI-based ThinApp package to deploy using the 'Software Installation' policy
3. Add the appropriate security groups and assign GPO to the appropriate OU's
4. Log into a machine/user specified by the GPO and confirm installation by looking in the Add/Remove applet of the Control Panel
5. Launch the application and test functionality

These scenarios demonstrate the portability of ThinApp packages and the two primary modes of execution.

Using ThinApp Assignments with VMware View

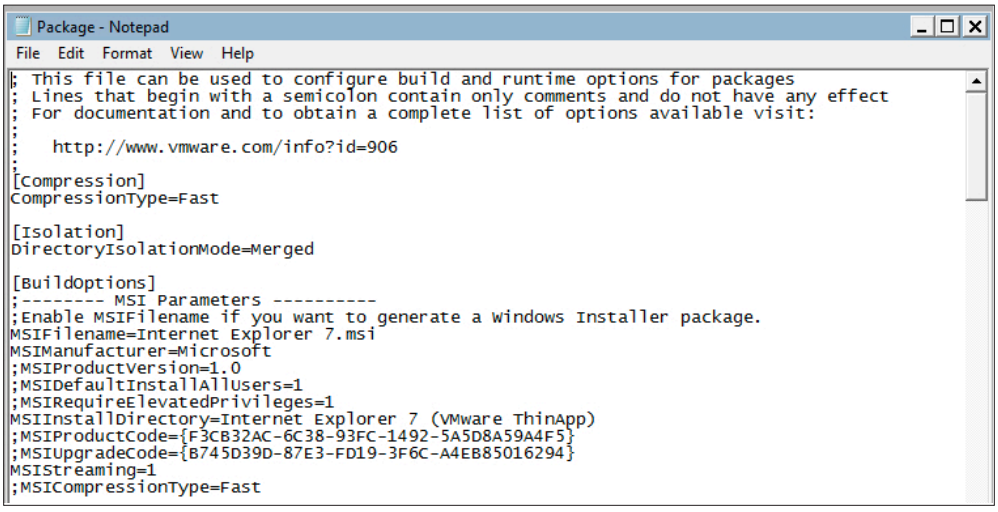
The VMware View Administrator allows ThinApp packages to be assigned to individual desktops or pools of desktops to allow for streamlined application deployment. The requirements to utilize this method are MSI-based ThinApp packages, a fileshare, and VMware View 4.5.

This video reviews the setup and operation of ThinApp Assignments in VMware View: http://www.youtube.com/watch?v=d19toUaelCw&feature=youtube_gdata_player

To Create the MSI-based Packages for Full and Streaming Deployment

ThinApp packages used for assignment through View Administrator must be in MSI format. The default selection of creating an MSI through the Setup Capture wizard will create a package that can be deployed only in Full (deployed) mode through View Administrator. To be able to deploy the package in Streaming mode as well, the MSIStreaming parameter must be set to 1. For this reason, when creating packages for a VMware View environment it is recommended to create packages using the MSIStreaming=1 parameter.

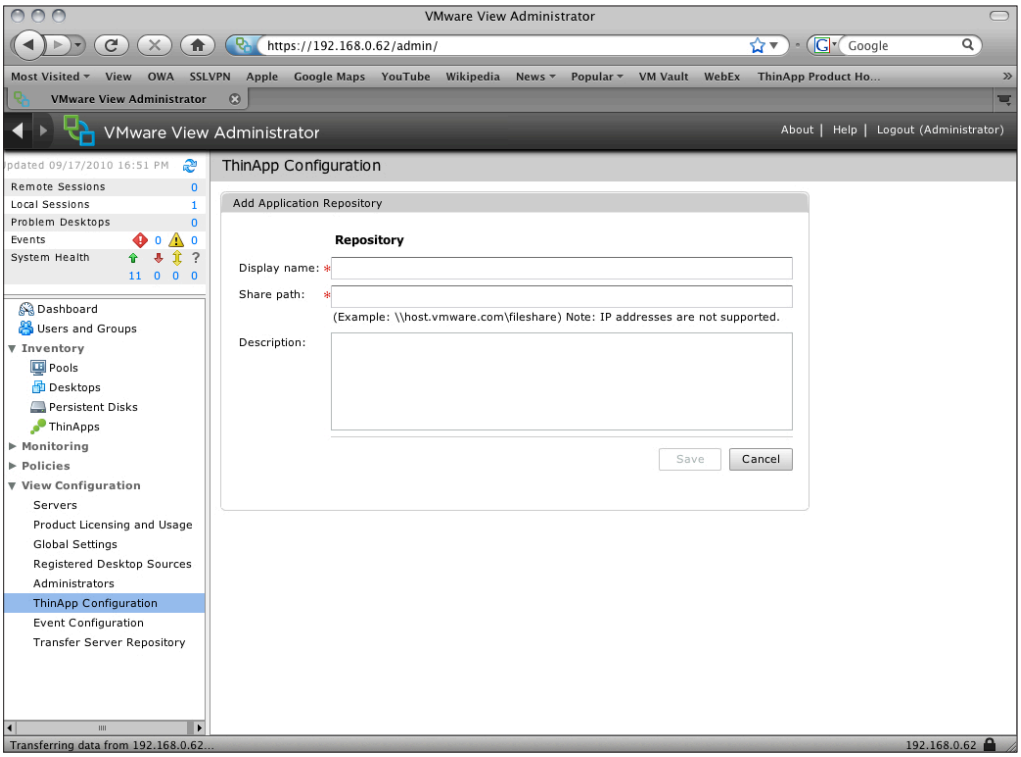
1. During the Setup Capture process edit the Package.ini MSIStreaming paremeter and set it to '1'.



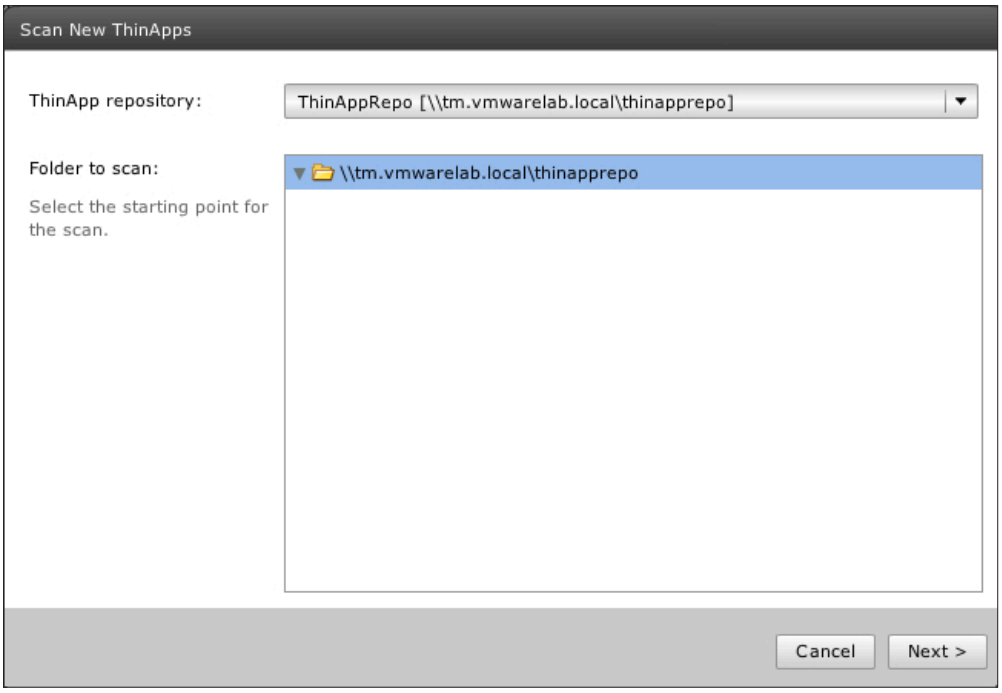
2. To utilize existing packages, edit the package.ini to enable MSIStreaming=1, then rebuild the package.
3. Copy all the files from the Bin directory to the fileshare that will serve as the ThinApp Repository. Share the repository so that View Administrator and end users can access. Consult the product documentation for specifics on security recommendations.

To Create the ThinApp Repository and Populate With ThinApp Packages

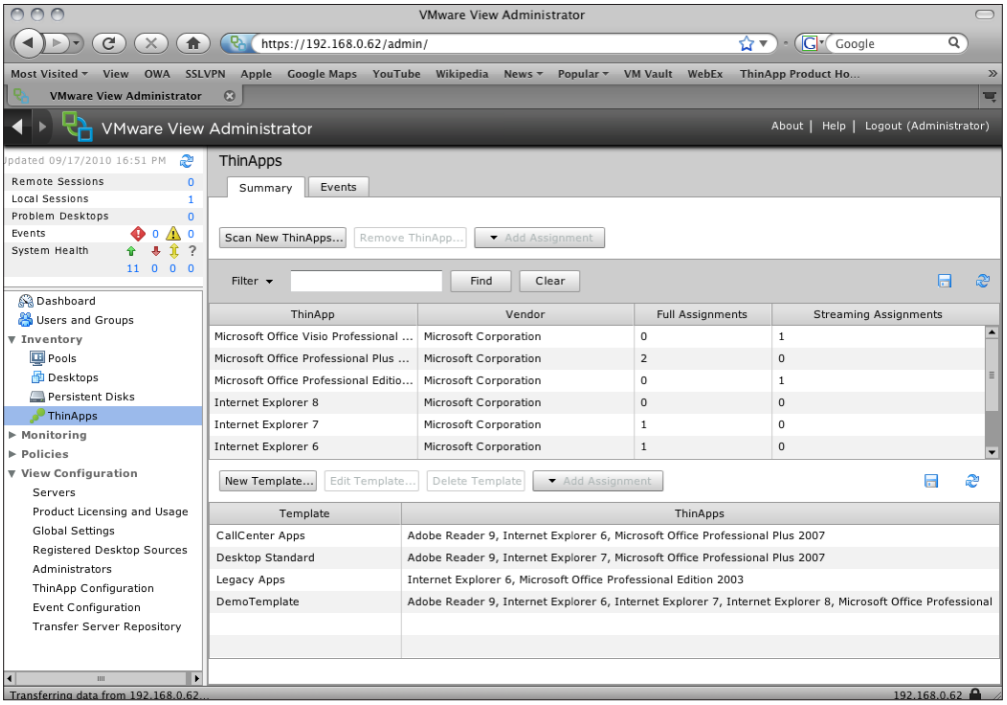
1. Add the ThinApp Repository specifying the appropriate path and optionally providing a description.



2. Navigate to the ThinApps configuration section and choose to 'Scan' for new applications, selecting the ThinApp Repository specified previously.



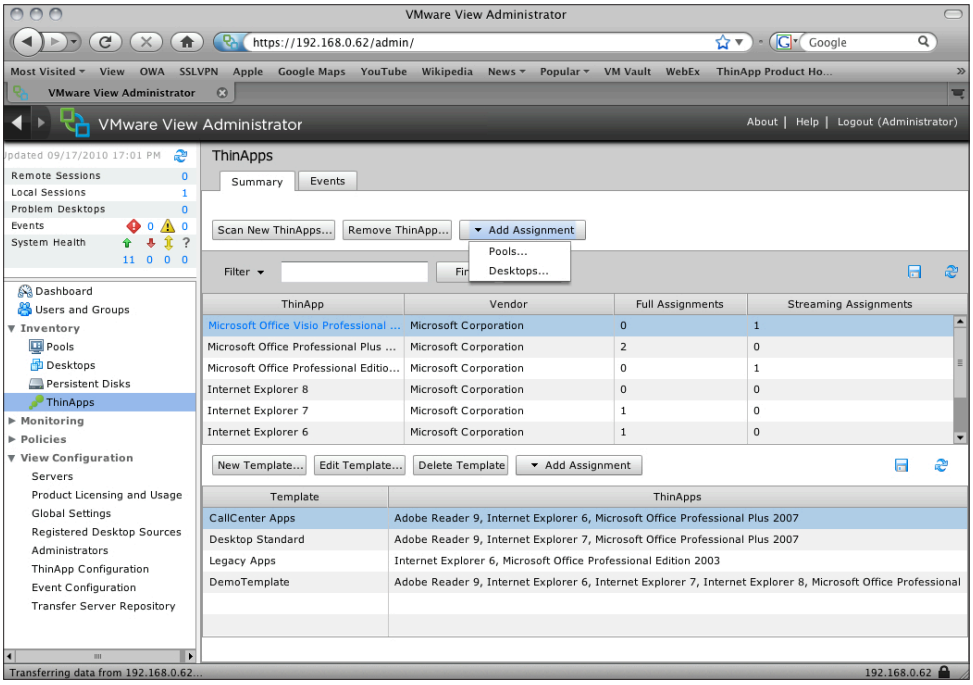
Select the MSI packages to add them to VMware View Administrator.



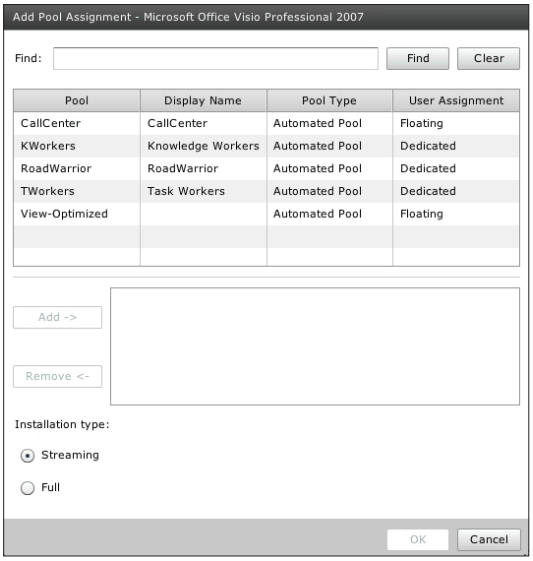
To Create ThinApp Assignments in View Administrator

1. (Optional) Create ThinApp templates which are groups of ThinApp packages. This allows you to make one assignment to a desktop or pool and deliver a group of applications instead of making multiple assignments. Click New Template, give it a name, and then add specific ThinApp packages to populate the template.

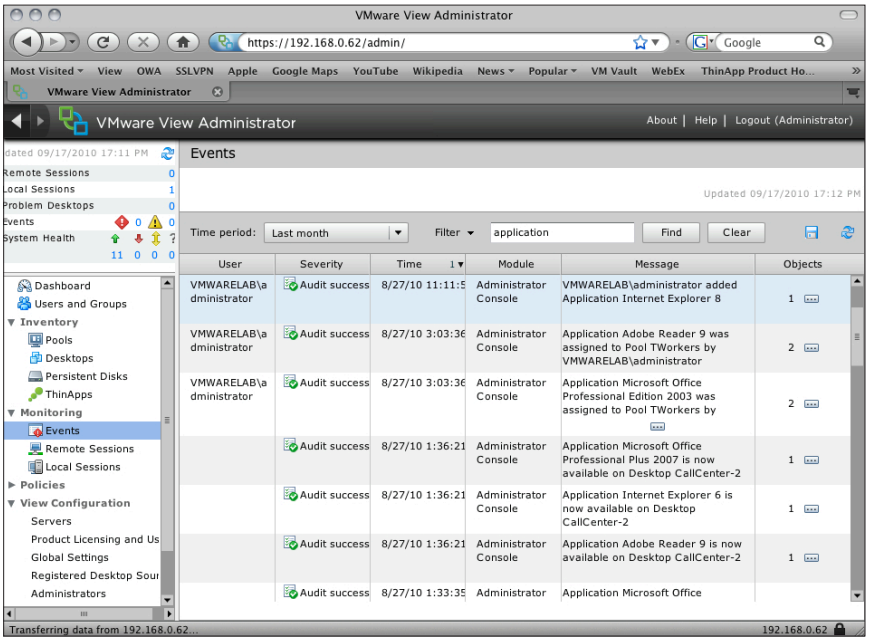
2. Select either a ThinApp Template or a single package and then click Add Assignment.



3. Select a single or multiple desktops or pools and then choose the Installation type.



4. Verify the Assignment and monitor progress by using the Events applet.



The process illustrated above reviews the process to set up a ThinApp repository and make ThinApp assignments within View Administrator. Application registration happens automatically when ThinApp packages are deployed in this manner. While not supported, it is possible to deploy the VMware View Agent onto physical machines and utilize ThinApp assignments for deploying to physical desktops.

To Utilize Script-based Application Registration

For deployment to non-View managed desktops, the use of script-based application registration can be automated to provide role-based access to ThinApp packages.

This video reviews the setup and operation of script based application registration: http://www.youtube.com/watch?v=PmPNqWnYv1U

The Application Registration Guide also reviews this process in detail and can be used as a reference for this activity. http://www.vmware.com/files/pdf/VMware-DG-ThinApp-AppRegistration-EN.pdf

1. Choose whether to utilize a local script or login script to run the ThinReg utility.

Create a script or edit an existing script to include the path to the fileshare hosting the ThinApp packages or to reference local directories that are populated with ThinApp packages.

For Example `%logonserver%\netlogon\thinreg.exe \\company.com\applications\*.exe`

2. Trigger the script by login or manually, and verify registration using the Add/Remove control panel applet or by checking File Type Associations.
3. Remove a user from the Active Directory authorized group and logoff/logon to see the application icon removed.
4. Attempt to launch the application and verify that access is denied.

To Utilize AppLink to Combine ThinApp Packages

Application Link is a feature that connects dependent application packages at runtime. This allows the administrator to build relationships between packages in order to package, deploy, and update component pieces in separate ThinApp packages rather than using Setup Capture to package all needed components into a single package.

Create links between packages for the following scenarios:

- Link runtime components, such as .NET, JRE, or ODBC drivers, with dependent applications. For example, you can link .NET to an application even if the local machine for the application does not allow for the installation of .NET or already has a different version of .NET.
- Package and deploy application-specific components and plug-ins separately from the base application. For example, you might separate Adobe Flash Player or Adobe Reader from a base Firefox application and link the components.

To Create an Application Link between Packages
Follow the process below to set up the link. You can also use nested links between multiple packages or create a link to a directory using a '*' wildcard to establish links to all components in that directory.

1. Create the package with the component that you want to link, build the package as an .exe and then rename the file to something other than an .exe extension to prevent users from running that package directly. (Optional) A .dat extension will be used for this example, AdobeFlashPlayer.Dat.
2. Create the capture of the originating package with the component already installed, for example, Mozilla Firefox.
3. In the originating package, Mozilla Firefox, open the package.ini file and edit the following line in the [Build Options] section:

   RequiredAppLinks=AdobeFlashPlayer.Dat
4. Place both packages in the same directory, locally or in the central fileshare.
5. Launch the Mozilla Firefox application and then navigate to the Adobe site to verify Flash functionality.

Note: Application Links can be specified as Required or Optional in the Package.ini. If specified as Required, the primary application will NOT launch if it cannot connect to the Linked application.

Updating ThinApp Packages

Providing application updates can be necessary to either provide additional application functionality for end users or to comply with administratively prescribed updates to software. When packaging applications, it is necessary to decide if the responsibility to package the application rests with the user or the administrator. Users who self-update virtualized applications will incorporate the application changes directly into the applications sandbox, which may increase the size significantly. If a user self-updates an application then those settings also may interfere with future updates provided by the administrator.

Packaging Updates and Modifications

There are three methods for capturing and deploying updates. Choose the method most appropriate for the update you wish to deploy.

Recapture

Recapture simply means going through the setup capture process again for the purpose of incorporating the updates in between the Setup Capture pre-scan and post-scan snapshots. The result of this process is a new package that has the new changes in configuration or updates embedded. For example, your original package was Microsoft OneNote. To create the updated package, simply install Microsoft OneNote and apply the most recent Service Pack, then build a new application package.

Sandbox Merge

This method consolidates updates from a sandbox into an existing project directory. To use the Sandbox Merge method, first launch the virtualized application onto a clean workstation. Then run the update, which will place the new files, registry, and configuration changes into the sandbox of that workstation. Then use the sbmerge utility provided by VMware to merge the changes from the sandbox of that capture machine into the existing project directory, and then rebuild the package to incorporate the changes.

Post-Capture

The post-capture method of incorporating updates involves manually placing folders in the appropriate directories of the capture, manually editing the registry files to include changes, and editing the package.ini file to change configuration settings. Use this method when you definitively know the files or registry changes that you want to make. This method does not require the use of the Setup Capture process but you must re-build the package using the build.bat file to incorporate the changes.

Deploying Updates

Once the applications updates have been incorporated into a package there are three methods for deployment.

Package Replacement

The package replacement method for updating application packages can be used for either streamed or deployed execution mode. If you have created an updated package and have an administrative window where no users will launch the applications then you can simply replace the original .exe based package with the updated .exe. Make sure that the filename stays exactly the same: users depend on the shortcuts previously created to launch the applications.

Side by Side Update

The side by side method for updating application packages can be used for either streamed or deployed execution mode. There is no requirement for application downtime. This method works by placing the new application package in the same directory as the original application package and incrementing the filename extension to an integer number. Subsequent updates can be placed in the directories with extensions .2, .3, and so on.

The implementation of this update strategy follows a simple process. When a user launches an application from a shortcut that references the original .exe, logic built into the package automatically checks for identical package filenames with a numeric extension in the same directory. If an updated package, such as Mozilla Firefox.2, is found, the application launches using the file with the highest numeric extension. Always keep the original .exe that is referenced by the shortcut in place as it is a necessary pointer for the application to launch with or without updated packages. There is no downtime for the users with this method of update and no change window required for the administrator. Users will execute the updated package as they launch the applications and the original application packaged .exe directs them to the updated package.

AppSync

Application Sync provides updates to unmanaged machines that connect over networks with some degree of latency. AppSync provides a mechanism for a differential transfer over http to the endpoint, thus it is only used for application packages in deployed execution mode. When an application starts, Application Sync can query a Web server or fileshare to see if an updated version of the package is available. If an update is available, the differences between the existing package and the new package are downloaded and used to construct an updated version of the package. The end user must have the rights to modify the packages. If not, then the

AppSync utility can be run as a scheduled service as a user with sufficient rights to perform the update. The updated package is then used for future launches of the application. Settings that configure the location for Application Sync and detailed configuration are contained in the package.ini file.

To Update ThinApp Packages Using the Side by Side Method

The Side by Side method of updating is very efficient and provides a built in fallback mechanism. The steps below will provide guidance on how to use the Side by Side method for packages on fileshares and can also be used for local packages.

This video demonstrates the Side by Side update method: http://download3.vmware.com/community/videos/Thinapp/ThinAppSidebySideUpdating.mp4

1. Create an initial package of an application that you wish to update. For example, Mozilla FireFox 3.0.2, built as a ThinApp package named Mozilla FireFox.exe
2. Create a second package of the updated version you wish to deploy. For example, Mozilla FireFox 3.0.6, built as a Thinapp package named Mozilla Firefox Update.exe
3. Copy the two packages into the same directory, either locally or on a fileshare.
4. Launch the initial package, Mozilla FireFox.exe and confirm the version, it is not necessary to close the application.
5. Rename the updated version package from Mozilla FireFox Update.exe to Mozilla FireFox.1.
6. Close the previous application or re-launch and verify the updated version.

To Utilize AppSync to Update ThinApp Packages

The AppSync functionality provides an easy to administer method of updating packages inside or outside the corporate network. The Application Sync feature is a setting that initiates the pull of a differential update package from a central http web service or UNC location. The interval for polling for updates and the location of the http service or fileshare is configurable along with other settings in the Package.ini file.

This video demonstrates using the AppSync functionality: http://download3.vmware.com/community/videos/Thinapp/AppSyncwithMultipleEntryPointsDemo.mp4

Optional: You can reuse the packages created in the previous section and skip to step 3.

1. Create an initial package of an application that you wish to update. For example, Mozilla FireFox 3.0.2, built as a ThinApp package named Mozilla FireFox.exe.
2. Create a second package of the updated version you wish to deploy. For example, Mozilla FireFox 3.0.6, built as a Thinapp package named Mozilla Firefox Update.exe.
3. Edit the Package.ini in initial package directory to include the AppSync parameter pointing to a URL or UNC location. Example syntax follows:
   ```
   AppSyncURL=https://<site.com>/<path>/<package_name>.exe
   AppSyncURL=file://<server>/<share>/<path>/<package_name>.exe
   ```
4. Deploy the initial package to a desktop.
5. Place the updated package in the URL or UNC location specifed in the AppSync parameter.
6. Launch the application. By default, the update should occur and pop up a message that the application has been updated.
7. Log out of the application and relaunch to confirm the update was successful.

Additional Resources

VMware ThinApp Technical Papers:

Reference Architecture, Streaming Information Guide, Deployment Guide, Application Registration Guide
http://www.vmware.com/resources/techresources/cat/91,158

VMware ThinApp YouTube Channel
http://www.youtube.com/vmwarethinapp

VMware ThinApp Blog

VMware ThinApp employees regularly post and participate on the VMware ThinApp blog site
http://blogs.vmware.com/thinapp/

VMware ThinApp Community
http://communities.vmware.com/community/vmtn/entdesk/thinapp

About The Author

Aaron Black is a Senior Technical Marketing Manager at VMware. In this role, his primary focus is to develop technical content to aid in the evaluation and implementation of VMware ThinApp technology. Aaron's background includes roles as a systems engineer and solutions consultant in the Technical Services organization. His previous positions include systems engineer with Citrix Systems, leading a technical corporate IT team at Sprint, and solutions design for customers of Choice Solutions, a platinum reseller of VMware products.

vmware®

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
Copyright © 2010 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Item No: VMW_10Q3_RG_ThinApp_EN_R1